AR-010-549

An Investigation into the Use of Web Publishing and Browsing in C3I Simulation

Andrew Pope

DSTO-GD-0181

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

An Investigation into the Use of Web Publishing and Browsing in C3I Simulation

Andrew Pope

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-GD-0181

ABSTRACT

With the advent of World Wide Web technology, and in particular web browsing, and its increasing use in Australian command support systems such as the Joint Command Support Environment (JCSE), it is essential that a capability within C3I (command, control, communications and intelligence) simulations such as the Distributed Interactive C3I Effectiveness (DICE) Simulation be developed. Over a period of three months, a suite of tools for reporting on simulation data, using a web browser as the basis for the user-interface, was developed. The end product is called WEBSTAR (WEB-based Simulation Tools for Analysis and Reporting). WEBSTAR is now in use within the DICE environment. A set of criteria for evaluating web applications is suggested. Insights into the issues associated with the use of web technologies in C3I simulation have been gained from the development of WEBSTAR and additional research. This has provided valuable information for future web technology research and development in DICE.

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR- 010-549
May 1998

APPROVED FOR PUBLIC RELEASE

An Investigation into the Use of Web Publishing and Browsing in C3I Simulation

Executive Summary

World Wide Web (WWW) technology, such as web publishing and browsing, has become very widespread in the civilian world. Similarly, command support systems (CSS) within the Australian Defence Force (ADF) are increasingly using web publishing and browsing as a means of communication.

The Joint Command Support Environment (JCSE), developed under the ADF's Joint Project 2030, is one CSS that includes facilities for operators to browse pages associated with other headquarters (HQ). Examples of information which could be disseminated using web browsing are concept of operations, high-level plans, intelligence reports and rules of engagement.

CSS form important components of a command, control, communications and intelligence (C3I) system. The Defence Science and Technology Organisation is working for the ADF to develop capabilities for modelling and simulation of C3I systems. Hence it is natural to conduct an investigation into the use of web publishing and browsing in C3I simulations.

An example application of this technology is where an HQ, equipped with a CSS (e.g. JCSE), provides web access to pages of external agencies. If one wished to train that HQ through the use of a simulation environment then that environment would need to support and maintain such external web pages through appropriate technologies.

A suite of tools for information retrieval from the Distributed Interactive C3I Effectiveness (DICE) simulation using a web browser as the basis for the user interface has been developed over a three month period. This suite is called WEBSTAR (Web-based Simulation Tools for Analysis and Reporting). It consists of a collection of forms for user input of information requests and web-browser based reports for information output showing the requested information.

There are a high number of reporting possibilities within the DICE simulation. Not all of these could be developed during three months. However, querying and reporting on simulation messages, detections of aircraft and aircraft take-offs and landings have been developed. Work is continuing on developing further reports.

A suggested set of criteria for evaluating web applications are: usability, functionality, flexibility, accessibility, response times, scalability, portability, robustness, and security. This may not be a complete list, but all of these are important criteria.

This report is not a complete account of all issues associated with the use of web publishing and browsing in C3I simulation. Rather, specific insights have been gained from developing WEBSTAR. These include:

- A web application takes longer to develop than an equivalent Local Area Network (LAN) application;
- Dynamically generated web pages can be created by Common Gateway Interface (CGI) programs, incorporating user input or database data;

- Installation and maintenance of client web application software is simple;
- One advantage of using a web application rather than a traditional LAN application is greater network accessibility. However, in order to maintain realism within the simulation, it is sometimes necessary to restrict information available to simulation players;

In order to decrease future development effort, code libraries should be developed from the code used in WEBSTAR, for retrieving and processing CGI parameters from Hypertext Markup Language (HTML) forms, and for outputting well-formed HTML markup.

If there is great uncertainty before application prototype development begins as to whether the prototype will be successfully demonstrated and hence developed into a full web application across the Internet, then perhaps the lowest risk strategy would be to use Local Area Network (LAN) Rapid Application Development (RAD) tools to develop the prototype. That way, if full development is rejected, then less cost will have been incurred than if the prototype had been developed as a web application. However, if it is likely that the prototype will be developed into a full web application across the Internet, then a web application prototype should be developed, then scaled up across the Internet.

Further research and development into web technologies in C3I simulation, especially WEBSTAR and DICE, should include:

- In order to maximise effectiveness of using various communications technologies, generic research should be conducted into selecting the most appropriate communications technology for particular tasks (e.g. is e-mail or web publishing most appropriate for the information being communicated?). Based on this research, write a report on the appropriate use of web publishing and browsing;
- Use the research presented herein into technologies to drive development of web browsing capabilities in DICE and to interoperate with JCSE.
- Incrementally develop more components to WEBSTAR (e.g. aircraft mission details);
- Perform algorithmic complexity analysis of WEBSTAR programs and thorough testing of WEBSTAR within many DICE scenarios.
- Allow sorting of WEBSTAR report records other than in chronological order of simulation time;
- Investigate web security;
- Investigate Java and Javascript for re-engineering of WEBSTAR;
- Investigate Dynamic HTML (both Microsoft and Netscape definitions);
- Develop a Web Peripheral Unit Interface (PUI) - a DICE PUI which receives messages from other agents and publishes HTML pages depending on the message contents;
- Incorporating ADFORMS->Text conversion into the Human Player (HP) facility, and subseqently incorporating this into the WEBSTAR message report;
- Adding a graphical facility to view the DICE simulation network;
- Investigate automatic simulation of standard ADF reports e.g. INTREP, MISREP with data retrieved from the simulation;
- Add ability of Human Player to publish own HTML pages easily;

WEBSTAR is now being used in the DICE environment. The development of WEBSTAR and additional research has provided valuable information for future web technology research and development in DICE.

Author

Andrew Pope
Information Technology Division

Andrew graduated with a B.Sc.(Hons) degree in Computer Science from the University of Leeds, England in 1992. After working as an I.T. contractor he joined the Defence Science and Technology Organisation's (DSTO) Information Technology Division (ITD) in 1996 as a Professional Officer within C3I Operational Analysis (C3IOA) Group. Andrew has since been involved in the Australian Army Command Support System (AUSTACSS) Defence Trial, Exercise Tandem Thrust and Exercise Pitch Black 96. He has also developed the Qualitative Data Analysis Tool (QDAT) software, for aiding analysis of textual data collected at trials and exercises.

Contents

Table of Figures

Abbreviations

AAP	Australian Associated Press
ADF	Australian Defence Force
ADFORM	Australian Defence Formatted Message
ADFORMS	Australian Defence Formatted Message System
ADSIM	Air Defence Study Interactive Model
C3I	Command, Control, Communications and Intelligence
CGI	Common Gateway Interface
COMMREP	Communications Report
CSS	Command Support System
DICE	Distributed Interactive C3I Effectiveness
DIO	Defence Intelligence Organisation
DSTO	Defence Science and Technology Organisation
GBR	Ground-based Radar
HQ	Headquarters
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
INTREP	Intelligence Report
INTSUM	Intelligence Summary
JCSE	Joint Command Support Environment
JORN	Jindalee Operational Radar Network
LAN	Local Area Network
LOG/PERS REPLEN	Logistics/Personnel Replenishment
MISREP	Mission Report
MIT	Massachusetts Institute of Technology
NIPRNET	Non-secure Internet Protocol Routing Network
OTHR	Over-the-horizon radar
PUI	Peripheral Unit Interface
RAD	Rapid Application Development
RCC	Regional Correlation Centre
SIPRNET	Secret Internet Protocol Routing Network
SITREP	Situation Report
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
URI	Universal Resource Indicator
W3C	World Wide Web Consortium
WAN	Wide Area Network
WEBSTAR	Web-based Simulation Tools for Analysis and Reporting
WWW	World Wide Web

1. Introduction

Browsing of published World Wide Web (WWW) pages on a web server using web-browsers has emerged as a very popular means of communication. The aim of this report is to discuss some of the issues involved with using web publishing and browsing in military simulation, when applied to Command, Control, Communications and Intelligence (C3I). Development of a Web-based Simulation Tool for Analysis and Reporting, called WEBSTAR, is used for illustration.

2. Why Use Web Publishing and Browsing in C3I Simulation?

Web publishing and browsing has become popular in the civilian world because of the advantages it offers over other means of communication. This report aims to justify the use of web technology in C3I system simulation by briefly considering these communications benefits and then addressing how the technology may be applied in the context of simulation.

2.1 Prevalence of Web Technology

In the corporate world, much work is being done on harnessing web technology in internets, intranets, and extranets to meet organisational goals. Similarly, the military is also seeking to utilise such technology to meet its goals. In the United States military, the Non-Secure Internet Protocol Routing Network (NIPRNET) and Secret Internet Protocol Routing Network (SIPRNET) are two networks over which web technology can be used. In Australia, the trend is for command support systems (CSS), such as the Joint Command Support Environment (JCSE), to increasingly utilise web technology.

2.2 Joint Command Support Environment

The JCSE has been established to provide the ADF with an integrated command support system applied in a military environment. JCSE operates at the strategic and operational level of command (i.e. the high-to-medium levels).

The JCSE is now designed to make use of the DEFSECNET digital network and provides service at up to SECRET level, including:
- e-mail;
- other message communication not using e-mail (e.g. Transmission Control Protocol/Internet Protocol [TCP/IP] sockets);
- web browsing, including browsing of imagery, Defence Intelligence Organisation (DIO) information and, in the future, newsfeeds from sources such as the Australian Associated Press (AAP).

The JCSE includes facilities for operators to browse pages associated with other headquarters. Examples of information which could be disseminated between ADF

agencies using web technology are concepts of operations, high-level plans, intelligence reports and rules of engagement.

Examples of information which could be obtained by ADF agencies from non-ADF agencies using web the JCSE web browser are weather and news reports. Weather information could be obtained through use of the Bureau of Meteorology web site and gopher pages, or other international agencies providing weather information. News could be obtained by browsing the AAP or Reuters web sites.

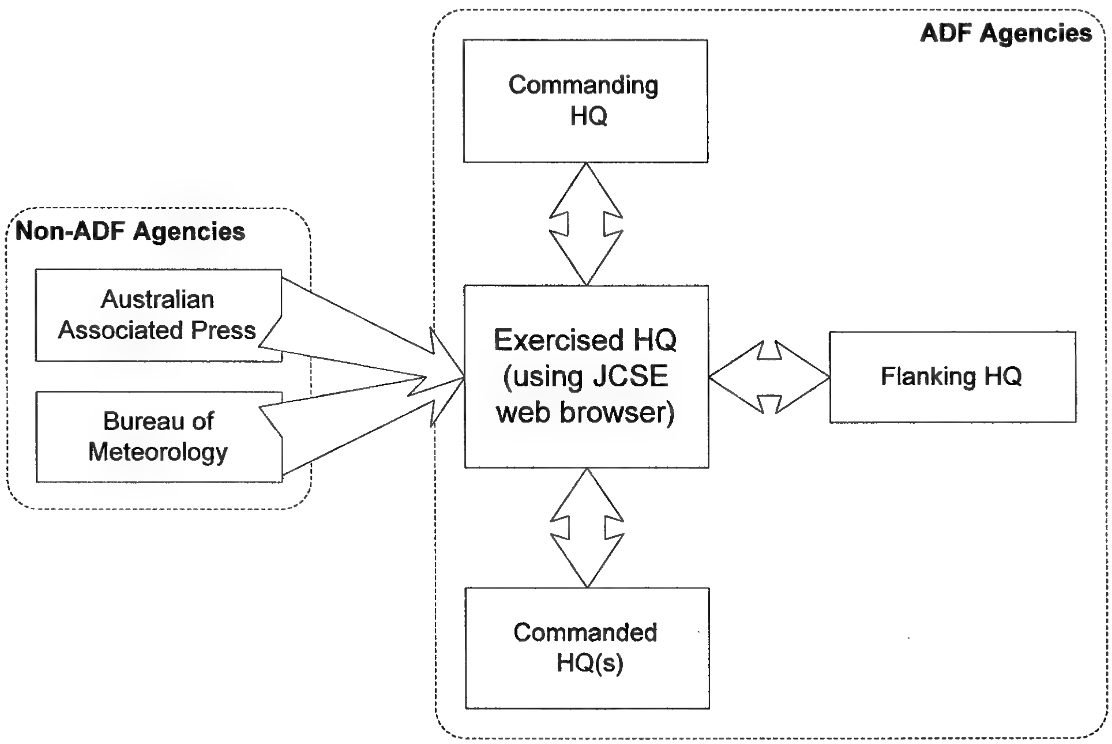

Figure 1 Using the JCSE Web Browser

2.3 Web Technologies and the DICE Simulation

CSS form important components of a command, control, communications and intelligence (C3I) system. The Defence Science and Technology Organisation (DSTO) is working for the Australian Defence Force (ADF) to develop modelling and simulation tools for studies of C3I systems. One tool under development is the Distributed Interactive C3I Effectiveness (DICE) simulation [Davies et al].

DICE is designed to stimulate and interact with other systems, including CSS (e.g. JCSE). The increasing use of web technologies in ADF CSS necessitates that the DICE simulation be further extended to cover this additional medium.

Figure 2 shows which means of communication various components of a DICE/JCSE simulation will use. The simulation consists of:

- several human players (HP), who are people playing the part of an ADF position;
- artificial agents, which simulate headquarters, units or personnel;
- one or more battle simulations, which provide stimuli for the simulation to which the players respond. For example, producing simulated aircraft flights detected by simulated radars. This could be done by the Air Defence Study Interactive Model (ADSIM) software, developed by DSTO [Haydon];
- a simulation controller, who as the name suggests, monitors and controls the simulation.

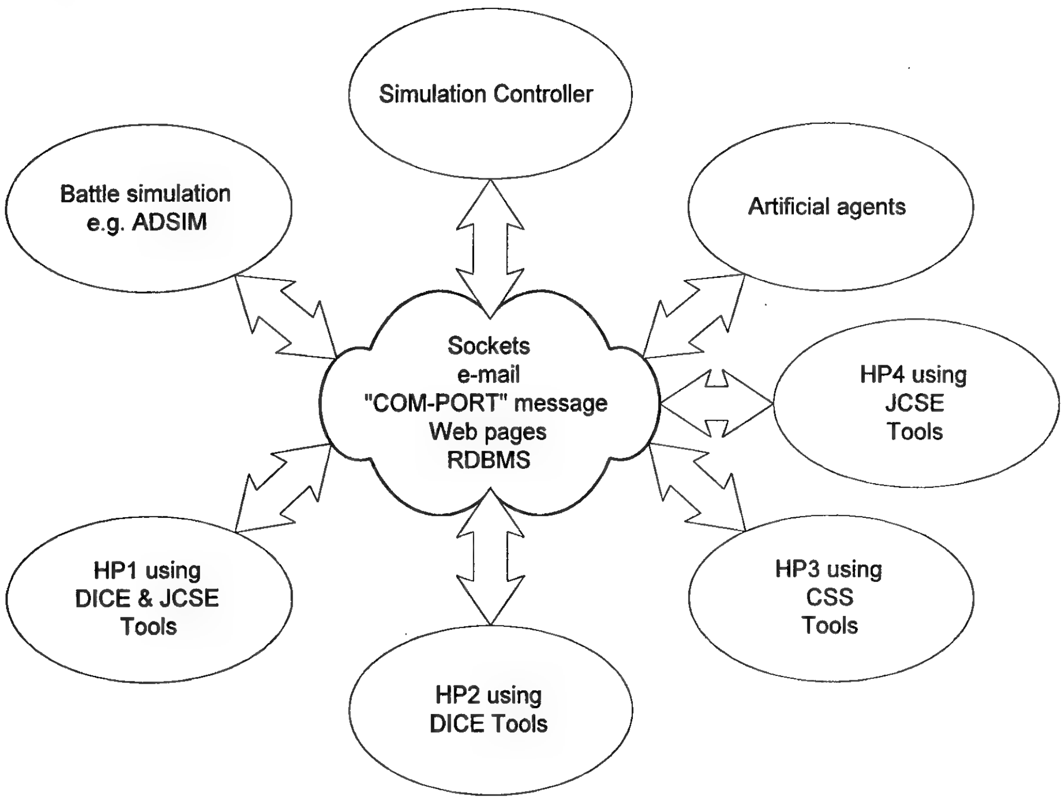

Figure 2 DICE/JCSE Communication

Each HP will use a selection of tools from DICE, JCSE and other CSS. JCSE tools include the Picture Manager, which shows a joint air, sea and land picture, message facilities (e.g. Lotus Notes Mail), and a web browser. DICE tools for the HP include WEBSTAR (Web-based Simulation Tools for Analysis and Reporting), the human player message facility (like a mail application), and a web browser. Where there is overlap between tools being used by a HP on one computer (e.g. if DICE and JCSE both have e-mail tools) then one only must be used. Other CSS have a variety of tools

which could be used in the simulation, including message facilities and battle map displays.

The JCSE/DICE communication simulation will employ different message communication modes, such as e-mail, messages employing the computer's COM port and a switch, TCP/IP sockets and relational database management systems (RDBMS). Web pages will also be used for communication.

2.4 DICE, WEBSTAR and ADSIM

The first work done in using web technology in DICE was to develop a Web-based based Simulation Tool for Analysis and Reporting, called WEBSTAR. The aims of this development were to:

1. Experiment with web technology in C3I simulation, in particular the DICE simulation.
2. Produce a useful software product. In particular, develop a web application for analysis and reporting in DICE to report on:
   - messages sent between simulation players (e.g. message from a fighter controller to a fighter aircraft to take off)
   - aircraft detections by radars (e.g. an F/A-18 fighter detected by a ground-based radar)
   - flights (e.g. a B52 took off at time 00:30:18).
3. Improve on current methods for retrieving information from DICE. This should include information retrieval both during simulation and post-simulation.
4. Add new information retrieval functionality to DICE.

WEBSTAR uses two data sources for input to its forms and reports:
- For message reporting, a database which stores all DICE messages passed in the simulation.
- For aircraft detections and flights, air-defence simulation log files updated by the air-picture simulation ADSIM [Haydon].

3. Development of WEBSTAR

3.1 Application

WEBSTAR is an application for simulation analysis and reporting. It uses a web-browser as the basis for its user-interface. The user-interface is designed to balance simplicity with functionality and to minimise waste of screen space (sometimes termed "real estate"). The user-interface consists of hyper-linked pages of three types:
- forms in which the user defines the data they are interested in and clicks a button to view the corresponding report;
- reports which show the data matching the user's request;
- on-line help.

The interface look is consistent over all forms and reports. All forms are accessible from any page in WEBSTAR.

3.1.1 Forms and Reports

WEBSTAR consists of three reporting components:
- aircraft flights
- aircraft detections
- messages.

These components are discussed in more detail in the following sections.

3.1.1.1 Aircraft Flights

This component is used to show the start and end times of aircraft flights. That is, the times when aircraft take off and land. These aircraft can be detected or undetected by the radars in the simulation. ADSIM provides the input data for aircraft flights.

An example of this form is viewing a summary of all aircraft in the area of interest. This can be used, in conjunction with the aircraft detections component, in post-simulation analysis to determine which aircraft were detected and which were not.

3.1.1.1.1 Aircraft Flights Form

The form shown in Figure 3 is operated by:
1. Entering into the form the range of times for which flights are of interest;
2. Clicking the "View Flights" button to view the report shown in Figure 4.

All times in WEBSTAR are in DICE simulation time, which starts at zero hours, zero minutes and zero seconds when the simulation starts. The user can specify Start Time and End Time, either or both of which may be empty. This leads to four combinations of Start Time and End Time:
1. If Start Time and End Time are both empty then all times are shown;
2. If Start Time is specified and End Time is empty then all times from the Start Time to the current time are shown;
3. If Start Time is empty and End Time is specified then all times before the End Time are shown;
4. If Start Time and End Time are both shown then all times between and including these times are shown.

There are no other entry fields on the Aircraft Flights Form. A button is provided, labelled "Reset all selections", whose purpose is to clear all field entries.

For example, if the user was interested in viewing all take-offs and landings during the first four-and-a-half minutes of simulation, then the form should be completed as shown in Figure 3.

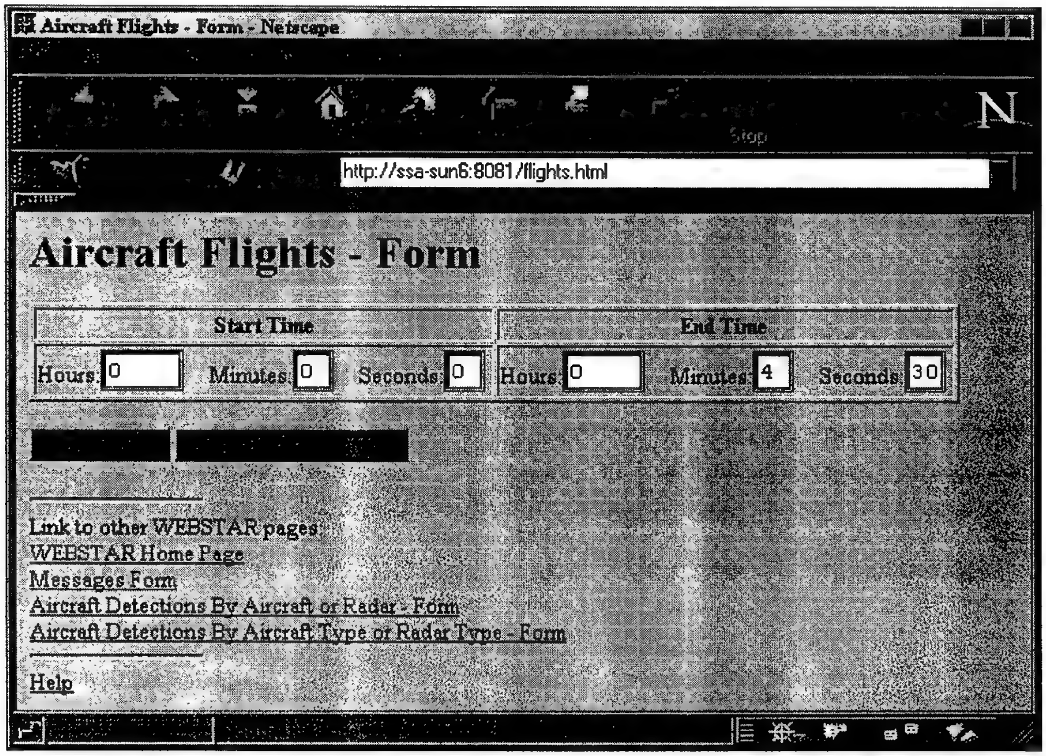

Figure 3 Aircraft Flights Form

3.1.1.1.2 Aircraft Flights Report

If the Aircraft Flights Form is completed as shown in Figure 3, and a particular simulation scenario is used to produce the relevant input data to WEBSTAR, then the Aircraft Flights Report shown in Figure 4 will be displayed in the user's browser window.

The Aircraft Flights report (and all other reports in WEBSTAR) shows, in a similar format to the form for consistency and speed of assimilation, what the user requested in the corresponding form, so that use of the browser "Back" button is not required to confirm for the user that the request is what they really wanted.

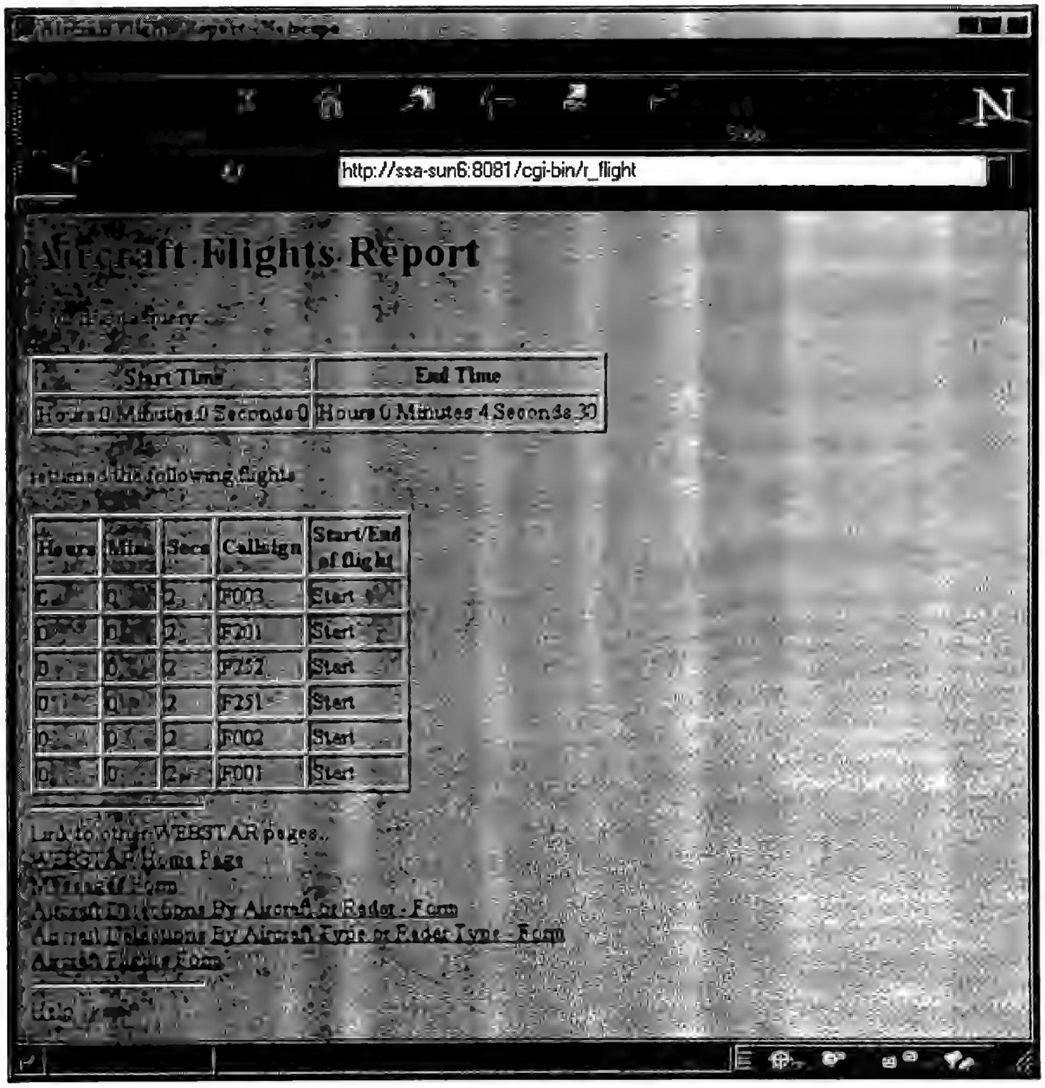

Figure 4 Aircraft Flights Report

The columns in the table are:

- Hours, Mins, Secs - the simulation time of the take-off/landing;
- Callsign - the callsign of the aircraft which took-off/landed;
- Start/End of flight - either Start (take-off) or End (landing).

The report in the example shows that six aircraft, with callsigns of F003, F201, F252, F251, F002 and F001 took off after two seconds of simulation. This is shown by the "Start" value the Start/End of flight column signifying the beginning of a flight (a take-off). No aircraft landed during the first four-and-a-half minutes of the simulation. If there were, this would be indicated by an "End" value in the Start/End of flight column.

3.1.1.2 Aircraft Detections

This component of WEBSTAR is for reporting on radar detections of aircraft in the area of interest. For example, viewing all detections of a particular aircraft by a ground-based radar.

There are three types of radar simulated by ADSIM, these being :
- Ground-Based Radar (e.g. GBR1, GBR6);
- Jindalee Operational Radar Network (e.g. JORN1, JORN3);
- Aircraft on-board radar, denoted by aircraft callsigns (e.g. F251 denotes the radar on board the aircraft with callsign F251).

There are two forms and two reports in the aircraft detections component. One pair of form/report is designed for specifying the detections of interest by the name of the detected aircraft and/or name of the detecting radar. The other pair of form/report is designed for specifying the detection of interest by the type of detected aircraft and/or the type of the detecting radar. The forms are deliberately separate, but could have been integrated (see section 3.2 for why this was done).

3.1.1.2.1 Aircraft Detections By Aircraft or Radar - Form

This form (and subsequent report) is used to view aircraft detections by specifying the detected aircraft or detecting radars by name (e.g. F252, F001 for aircraft and GBR2, JORN1, F252 for radars).

With the form the user can specify:
- times, in the same way as in the Aircraft Flights Form previously described in section 3.1.1.1.1;
- the aircraft detected by the radar;
- the radar(s) which detected the aircraft.

An additional radio button can be used to specify AND/OR conditions. This affects the detections shown in the report in the following way.

- If no aircraft are selected and/or no radars are selected, then this button is irrelevant. It does not affect the detections reported.
- If there is *at least* one aircraft selected, and there is *at least* one radar selected, then this radio button affects the information output in the report, as follows:

    1. If AND is selected, then the detections in the report will have one of the aircraft selected as the aircraft detected, *and* one of the radars selected as the radar which detected the aircraft.
    2. If OR is selected, then the detections in the report will have one of the aircraft selected as the aircraft detected, *or* one of the radars selected as the radar which detected any aircraft (not necessarily one of those selected).

An example form is shown in Figure 5 and an example report is shown in Figure 6. The report shows the detections occurring in the first two minutes of the simulation of aircraft with callsigns F003 and F201, AND detected by ground-based radar GBR6.

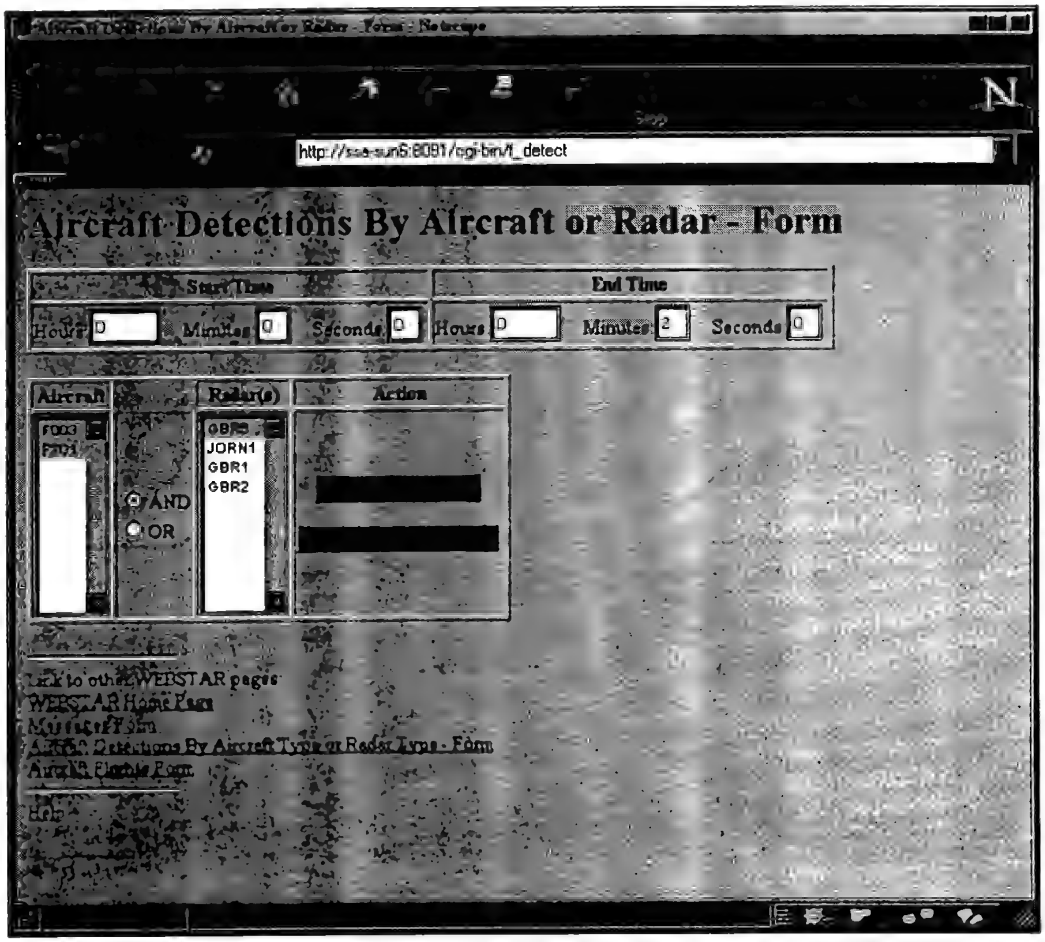

Figure 5 Aircraft Detections By Aircraft or Radar - Form

3.1.1.2.2 Aircraft Detections By Aircraft or Radar - Report

The report shows the user's request, in a similar format to the form, and the detections returned by WEBSTAR. The columns in the table are:

- Hours, Mins, Secs - the simulation time of the detection;
- Detected_AC_Num - the callsign of the aircraft detected by the radar;
- Detected_AC_Type - the type of aircraft detected by the radar;
- Detected_By - the name of the radar which detected the radar;
- Detected_By_Radar_Type - the type of radar which detected the radar;
- Lat (degs) - the latitude of the aircraft when detected (in degrees);

- Long (degs) - the longitude of the aircraft when detected (in degrees);
- Bearing (degs) - the bearing the aircraft was heading when detected (in degrees);
- Altitude (feet) - the altitude of the aircraft when detected (in feet).

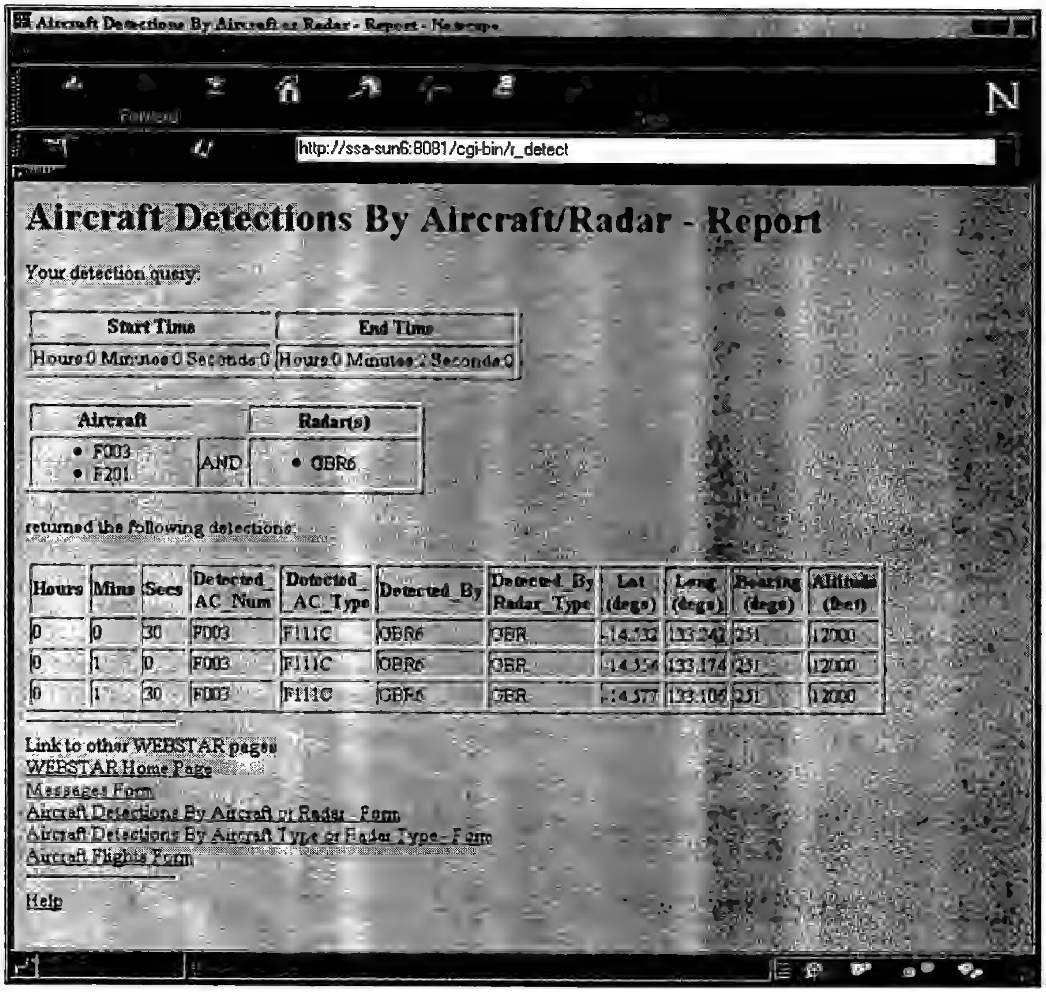

Figure 6 Aircraft Detections By Aircraft or Radar - Report

3.1.1.2.3 Aircraft Detections By Aircraft Type or Radar Type - Form

This form is used to view aircraft detections by specifying the type(s) of aircraft detected and/or the type(s) of radar which detected the aircraft (e.g. B52-G, F/A-18 for aircraft types and GBR, JORN for radar types). With the form the user can specify:

- times, in the same way as in the Aircraft Flights Form previously described in section 3.1.1.1.1;
- type of aircraft detected (There are seventeen types of aircraft simulated by ADSIM, e.g. F-111C, F/A-18, B52-G);
- type of radar which detected the aircraft;

An additional radio button can be used to specify AND/OR conditions as described in section 3.1.1.2.1.
An example use of the form is shown in Figure 7. The user wants to view the detections which occurred during the first two minutes of the simulation, of all F111C aircraft, detected by all ground-based radars. The corresponding report is shown in Figure 8.

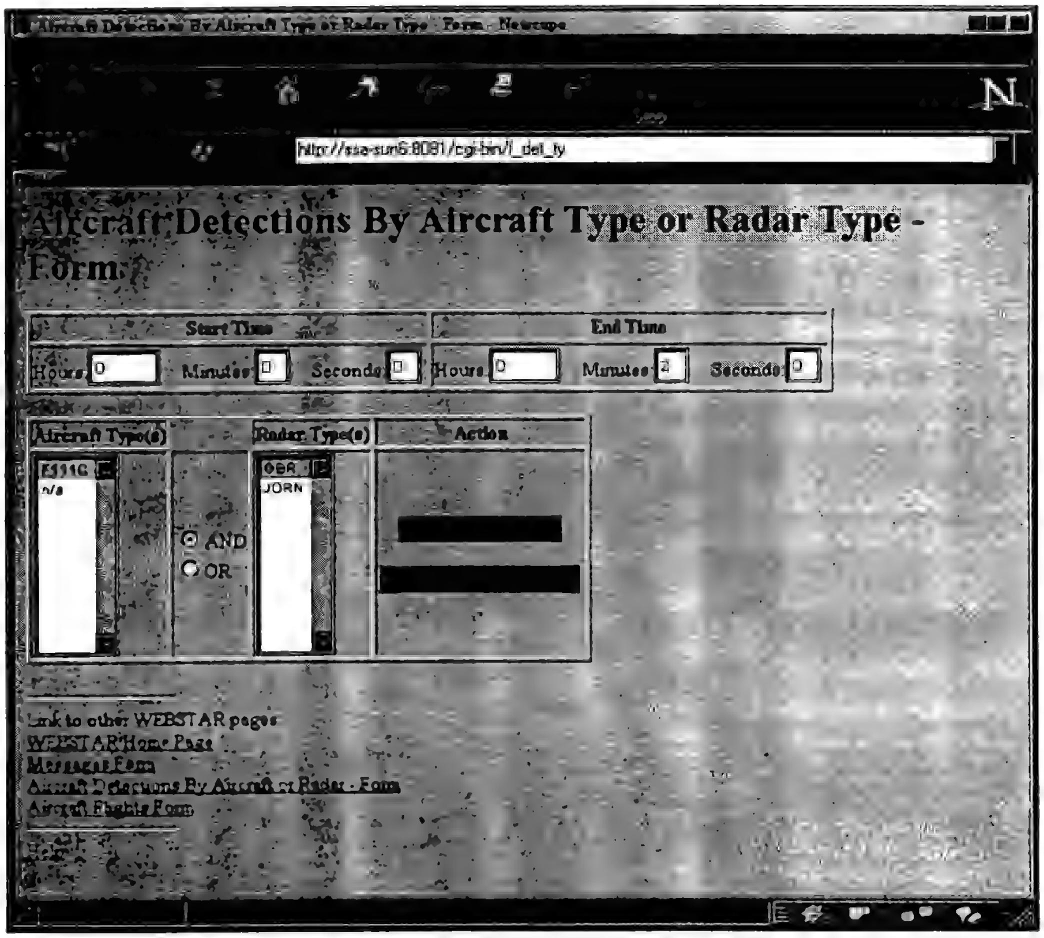

Figure 7 Aircraft Detections By Aircraft Type or Radar Type - Form

3.1.1.2.4 Aircraft Detections By Aircraft Type or Radar Type - Report

The report shows the user's request, in a similar format to the form, and the detections returned by WEBSTAR. The columns in the table are:

- Hours, Mins, Secs - the simulation time of the detection;
- Detected_AC_Num - the callsign of the aircraft detected by the radar;
- Detected_AC_Type - the type of aircraft detected by the radar;
- Detected_By - the name of the radar which detected the radar;

- Detected_By_Radar_Type - the type of radar which detected the radar;
- Lat (degs) - the latitude of the aircraft when detected (in degrees);
- Long (degs) - the longitude of the aircraft when detected (in degrees);
- Bearing (degs) - the bearing the aircraft was heading when detected (in degrees);
- Altitude (feet) - the altitude of the aircraft when detected (in feet).

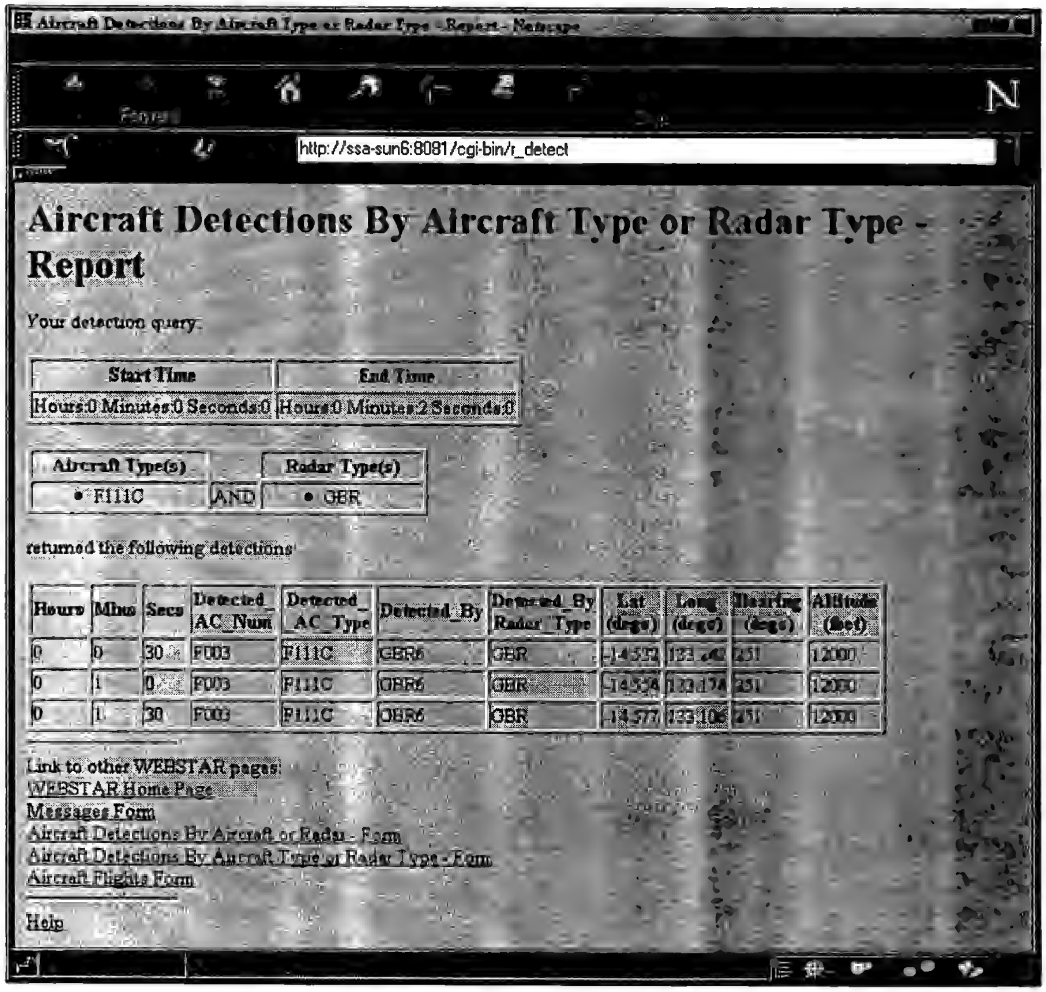

Figure 8 Aircraft Detections By Aircraft Type or Radar Type - Report

3.1.1.3 Messages

This component of WEBSTAR is for reporting on messages between nodes in the DICE simulation. For example, viewing all messages between an over-the-horizon radar (OTHR) and an air-picture fusion unit, the Regional Correlation Centre (RCC).

3.1.1.3.1 Messages Form

With the form the user can specify:

- times, in the same way as in the Aircraft Flights Form previously described in section 3.1.1.1.1;
- the sender of the message(s);
- the recipient of the message(s);
- an AND/OR radio button as described in section 3.1.1.2.1.

An example of use of the form is shown in Figure 9 and an example report is shown in Figure 10. The example is viewing all messages between the OTHR and the RCC (that is, sender is OTHR and RCC is the recipient) during the first five minutes of simulation.

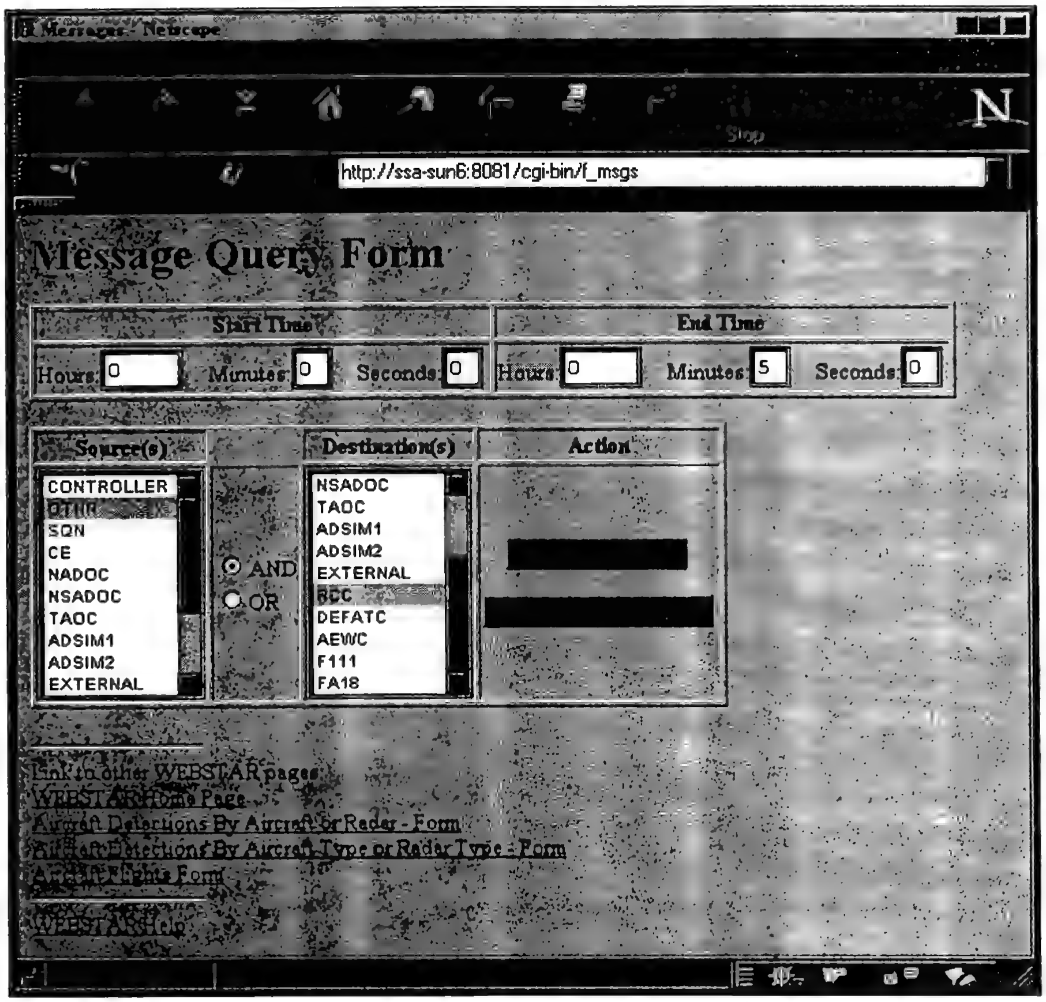

Figure 9 Messages Form

3.1.1.3.2 Messages Report

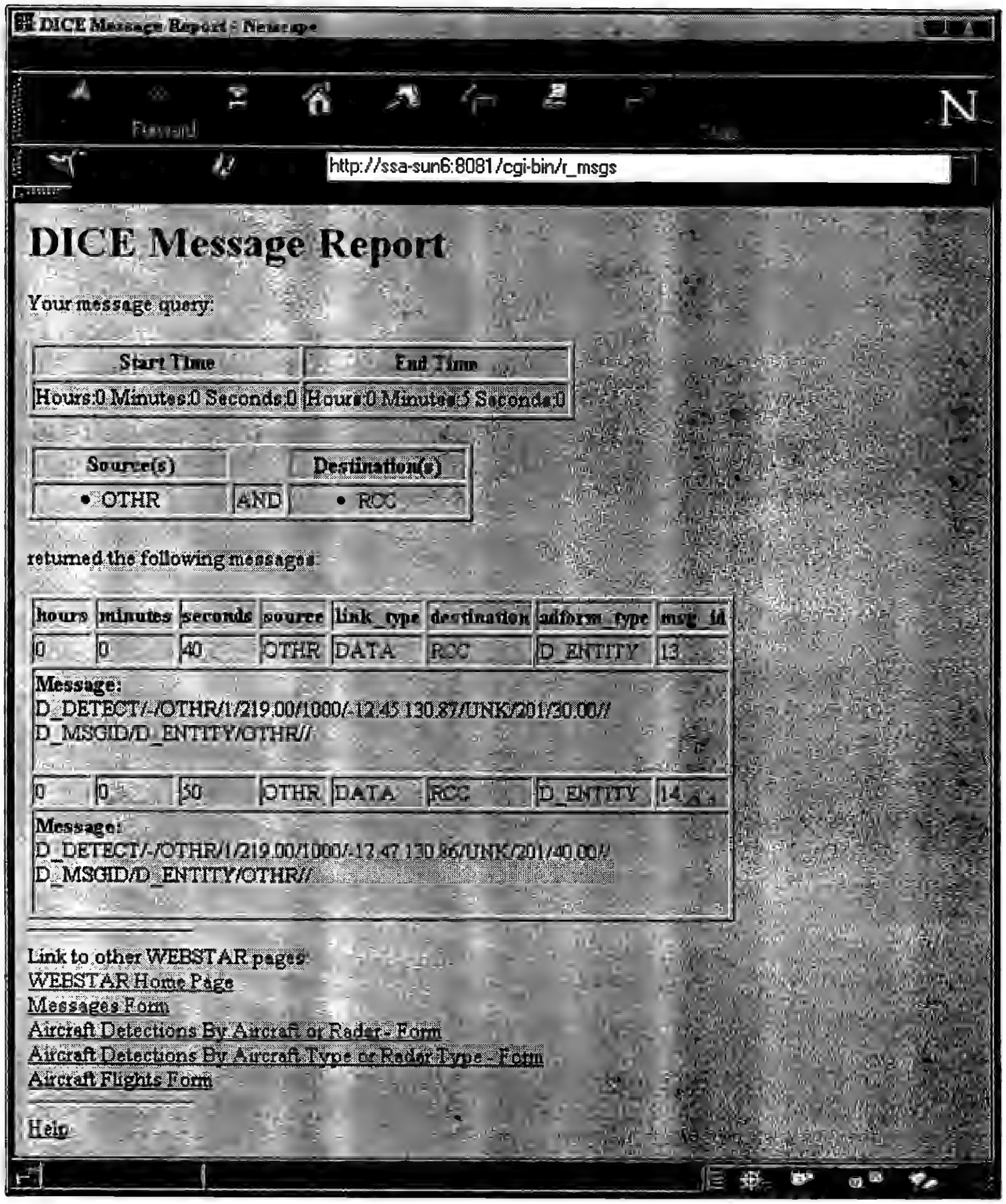

Figure 10 Message Report

The report shows the user request, in a similar format to the form, and information on the message, as well as the message content. The message is shown in slash-delimited Australian Defence Formatted Message System (ADFORMS) format. This is a standard for messages in the ADF.

The columns in the table have the following meaning:
- hours, minutes, seconds - the time the message was sent (in simulation time);
- source - the sender of the message, which can be any node in the simulation;
- link_type - the type of communication bearer used for the message;
- destination - the recipient of the message, which can be any node in the simulation;
- adform_type - the type of ADFORM sent;
- msg_id - a unique identifier for the message.

3.1.2 On-Line Help

There is an on-line help facility in WEBSTAR, which is also viewed in the web-browser window.

3.2 Features

WEBSTAR meets the aims described in section 2.4 for the following reasons. It is an improvement on current methods of information retrieval, and has added information retrieval functionality because:
- Information is reported in an easily assimilable form.
- A consistent user interface appearance and functionality is maintained throughout the application.
- The user interface balances simplicity with functionality. For example, rather than integrate the two aircraft detection forms together and add more AND/OR radio buttons, which could confuse the user, the forms are split into two separate forms.
- All users on the network can now easily access data using custom-built information retrieval tools (WEBSTAR). Previously, message data could only be accessed at one computer, that of the simulation controller.
- The tools allow users to focus on specific detection data from a large data set, by selecting the data of interest on the forms.
- There is greater flexibility in message querying (the user can now search on time and combine senders and recipients using the AND/OR radio button) rather than searching by senders OR recipients only, as previously.
- Forms and reports can be added to WEBSTAR, re-using much of the code from existing components.

Information can be retrieved from DICE both during simulation and post-simulation, using the same user -interface. The concept of web-browser based simulation tools for analysis and reporting has been demonstrated. More components are now being developed and WEBSTAR is now in use in the DICE simulation.

3.2.1 Usability

WEBSTAR has a consistent appearance and functionality throughout. The user-interface balances simplicity with functionality. Each form has a similar layout, using

HTML forms and tables, and the function of each form is very similar, so that the user's model of how the application works matches closely how it actually works [Cooper].

Every form has Start Time and End Time fields, a "View Report" button for displaying the report, a Reset button for resetting the selections and entries on the form, and links to all other forms and the WEBSTAR home page. There are also links to related on-line help pages. All forms are available via links from every form and report in WEBSTAR.

On-line help contributes greatly to usability. It has screen shots of forms and reports and descriptions of the fields on the forms and reports. There are example uses of each form, along with accompanying report examples.

3.2.2 Functionality

WEBSTAR provides four forms and reports, from a very high number of reporting possibilities. However, when new requirements are found, new forms and reports can be added to WEBSTAR, re-using much of the code from existing components.

There is greater flexibility in message querying in WEBSTAR than in the previous message viewing tool. This tool was only available on one computer (the simulation controller's machine). Now message information is available to any user on the network with a web browser. However, see the note in section 3.2.4.

The user can now search on time and combine senders and recipients using the AND/OR radio button, rather than searching by senders OR recipients only, as previously. Messages to/from mutiple sender(s)/recipient(s) can be viewed in the one WEBSTAR report.

The tools within WEBSTAR allow users to now focus on specific detection data from a large data set. There is now the ability to specify times, aircraft, radars, aircraft types and radar types of interest.

3.2.3 Flexibility

WEBSTAR uses *dynamically generated* HTML forms and reports. That is, they are created by programs which interrogate data sources to get content for the form or report. The difference between *static* and *dynamic* web pages is that static pages are stored on a file system and do not change, whereas dynamic pages do not 'exist' as such, not being stored on a file system. The programs create the HTML page 'on-the-fly'.

Dynamic forms and reports show the latest information available and can be used on different simulation scenarios without having to edit the HTML for the page. Hence dynamic forms and reports are more flexible than those that are static.

3.2.4 Accessibility

WEBSTAR is highly *accessible* since it can be used on any computer on the network with a suitable web-browser. The network can be either a closed network, behind a firewall, physically isolated from any other network, or open (for example, the internet).

Note that in order to maintain realism during a simulation, it is sometimes necessary to restrict a player's access to information. This remains to be investigated and incorporated into WEBSTAR.

It could be argued that the Ingres interactive Structured Query Language (SQL) terminal monitor, ISQL, could be used for querying and reporting, but in my experience, composing SQL is not something which the general user wants to do. Another alternative is a graphical query tool for querying, such as that used in Microsoft Access. This does not provide customised, easy access to the data which the user most frequently requires. WEBSTAR enables customised, easy access to the most frequently used data (from a human player's perspective) in DICE.

WEBSTAR is also accessible because *no installation is required* on most computers, since most computers have web software already installed. If not, all that is required is a web browser, TCP/IP software and a physical connection to the DICE network.

3.2.5 Response Times

Response times will depend on the algorithmic complexity of the WEBSTAR programs, delays caused by file locking and database locking, the load of the server(s) hosting the HTTP servers, network speed, the size of the input data and the type and number of processes involved in database locking.

3.2.5.1 Algorithmic Complexity
While some testing of WEBSTAR with DICE simulation scenarios has been done, no formal algorithmic analysis has been completed on WEBSTAR. This needs to be done.

3.2.5.2 File Locking
All WEBSTAR access to EVENTS.DAT is read-only, so there is no delay produced by other processes having to wait for WEBSTAR to finish accessing EVENTS.DAT. The only other processes which access EVENTS.DAT are ADSIM itself, which writes to it, and the ADSIM PUI, which reads it. Thus no locking conflicts can occur between two processes, since only one (ADSIM) needs write access.

3.2.5.3 Database Locking
The Messages Form and Messages Report access the DICE Ingres database, using SQL statements embedded in the host language C. Only those parts of the database relevant to messages, simulation nodes and communication links are accessed. Many other concurrent processes access the same tables. Hence the Ingres database must enforce a

locking mechanism to maintain database integrity. WEBSTAR was developed with concurrency in mind, and that it should not alter the performance of the DICE simulation noticeably.

3.2.5.3.1 Performance Degradation of DICE

Both WEBSTAR programs which access the DICE database (the Messages Form and Messages Report) issue only SQL SELECT queries, which gain read (share) locks on the underlying tables via the Ingres locking system. Because share locks force other DICE processes wishing to update data to wait until the shared lock transaction is completed, this will degrade the performance of DICE.

If the user has made a message request which results in a large number of messages being processed by the WEBSTAR program, this delay may be large enough to slow the other DICE processes to adversely affect the simulation. Whether this occurs depends on which processes need to access the message tables locked by WEBSTAR, and what data is required for update. Testing of WEBSTAR has confirmed that this has not thus far been a significant problem, for the particular simulation scenarios of interest.

3.2.5.3.2 Deadlock

All processes in DICE and WEBSTAR that access the database have error-handling code for gracefully dealing with deadlock.

3.2.5.4 HTTP (Web) Server CGI TimeOut Limit

Web servers can be configured to timeout after a specified time when waiting for a CGI program to terminate. This value must, of course, be long enough to allow most processing to complete, but short enough to not delay other processes and therefore slow the simulation. Only exceptionally long program runs should be halted. However, within DICE, short delays to processes of only a few seconds will slow the simulation. This is not an adequate method of controlling database locking, but is useful for stopping wayward programs and limiting wait times.

3.2.5.5 Effect on Server Host Machine Load

The WEBSTAR programs do not exert a great load on the Web server host machines. The load has not yet been sufficient to slow the DICE simulation significantly, having run WEBSTAR with some DICE scenarios. Thorough testing needs to be conducted.

3.2.5.6 Network Speed

The speed and load of the network between the web client (the web browser) and the web server will, of course affect the speed of response of the server to client requests. DICE simulations have been run on a small and large network, and response times have been adequate.

3.2.5.7 Size of Input Data

Because WEBSTAR must read every line of the input data file from ADSIM, its size will affect the performance of the aircraft detection and aircraft flights forms and

reports. The performance of the messages form and report depends on the size of the database tables in the DICE database. It is not anticipated that users will retrieve such large amounts of data, because this makes getting useful information from the data difficult. However, the user may not know that a large amount of data matches the request before the request is submitted. Retrieval of one-hundred or greater messages will produce a response time of more than thirty seconds. Very large retrievals will be halted by the web server timeout (see section 3.2.5.4).

3.2.6 Scalability

HTTP enables scalability because of its distributed nature - the service of different pages can be performed by different machines. That is, the HTTP server load can be spread over many computers, with the limitation of one page being serviced by at most one computer. WEBSTAR is built on HTTP and so can be built upon in a distributed fashion. For example, if one computer became overloaded, some WEBSTAR forms and reports could be moved to another computer, decreasing the load on the overloaded computer.

3.2.7 Portability

The client side of WEBSTAR is *portable* since web-browsers which support HTML forms are available for PC, UNIX and Macintosh operating systems.

The server side of WEBSTAR can be fully implemented on UNIX, and 75% of it can be implemented on Windows 95. The message component uses a UNIX CA-OpenIngres database, which is only accessible from executing a UNIX executable program. See section 3.3 for more information on the underlying software architecture of WEBSTAR.

3.2.8 Robustness

At the time of writing, WEBSTAR has very little user input validation. Robustness partly depends on the robustness of the web-browser itself. If it is unstable, then so WEBSTAR will be. WEBSTAR has been tested on UNIX and Windows 95 Netscape Navigator 4.0 and on Internet Explorer 3.02.

3.2.9 Security

DICE operates at the security classification of the underlying network. DICE does not add any security. For example, if the computer network through which DICE simulation nodes are connected is classified as UNCLASSIFIED, then DICE may process UNCLASSIFIED simulation data. Likewise, if the network is classified up to SECRET level, then DICE may process up to SECRET simulation data. The data in the simulation must match the classification of the underlying network.

Hence, when connected to the ADF's DEFSECNET network, (for instance to connect to JCSE), DICE will operate at the SECRET level.

During the development of WEBSTAR, security of the application, and its environment, was analysed. The following issues regarding security were determined.

Because WEBSTAR is a web application, the same security issues associated with the Internet apply to WEBSTAR. However, special mention should be made of CGI security. There are security issues in writing CGI executables [Van Biesbrouck].

If the CA-OpenIngres Spyglass HTTP (web) server is run as the `dice` user, any user of web-browsing software with network access to the web server can access the DICE Ingres database as the `dice` user. That is, any user may gain database access privileges of the `dice` user. This subverts the normal Ingres security constraints which would apply when accessing Ingres databases [Computer Associates].

For other information on security within WEBSTAR, see documentation on UNIX operating system security, Windows 95 operating system and HTTP server security [Sambar] and [Computer Associates].

3.3 Underlying Software Architecture

WEBSTAR is based on HTTP, which uses the client-server model. That is, the client requests data from the server and the server responds to the request with data.

3.3.1 CGI, HTML and HTTP

One technology which enables the servicing of client requests by sending back responses is the combination of a Common Gateway Interface (CGI) program and a Hyper-Text Transfer Protocol (HTTP) server. The client requests must also be in HTTP. HTTP provides servicing of many types of file (known as Multimedia media types - [Borenstein]), the most common of which is HTML [Berners-Lee 1995]. WEBSTAR uses CGI, HTTP and HTML.

3.3.1.1 Comparison of Dynamically Generated HTML with Statically Generated HTML

CGI works in tandem with an HTTP server to service requests from web-browsers for dynamically generated HTML. That is, HTML that changes frequently over time. CGI programs create HTML 'on-the-fly' and this HTML is not stored in a file. Compare this with statically generated HTML, which is a static page stored in a file.

3.3.1.2 How Does CGI Work With HTML?

The CGI defines a set of environment variables which contain information about the client's request. These are made available by the HTTP server to the CGI program, which uses them in composing its HTML response. A CGI program and HTTP server service client requests from a human reader as follows, shown also in Figure 11:

1. An HTTP connection between the server and client is created.
2. The server uses its internal configuration to answer the following question. Is the request for a statically generated HTML file or a dynamically generated HTML page? If the page is a static page stored in a file, then the server sends the file requested by the browser. If the page is a dynamically created (CGI) page, then the HTTP request is passed on to the corresponding CGI program via the CGI environment variables.
3. The CGI program receives the request and accesses the environment variables to obtain information on the request. It can also access any parameters from the request via the standard input device (stdin) when the user request uses the HTTP POST method [Berners-Lee 1995].
4. Information is retrieved from the simulation and incorporated into the HTML-formatted dynamically created HTML page.
5. The HTML dynamically created by the CGI program is output to the standard output device.
6. The HTTP server takes the dynamically generated HTML and compiles it into an HTTP-formatted response to the browser.
7. The request is complete and the HTTP client-server connection is terminated.

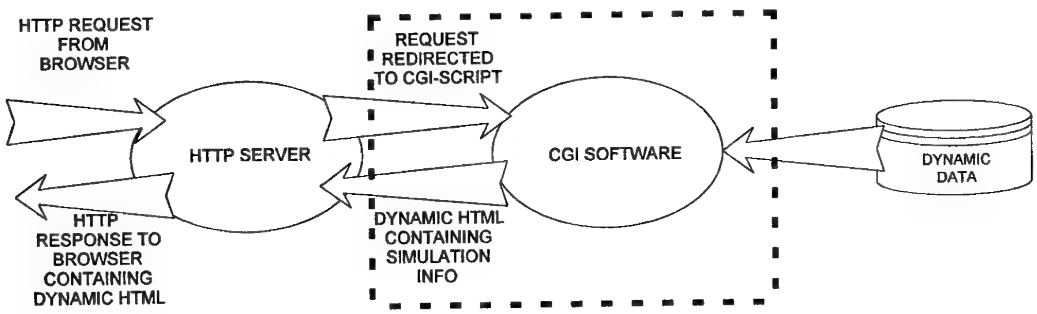

Figure 11 Generic CGI Software Environment and Operation

3.3.2 WEBSTAR Data Sources - DICE and ADSIM

As shown in Figure 11, CGI programs must get their data from some data source. WEBSTAR will retrieve its data from the DICE Ingres Database and the ADSIM data files.

DICE runs on UNIX under the Solaris Operating System. Hence most DICE components run on UNIX. DICE uses a CA-OpenIngres database, running on UNIX, to store its messages. ADSIM runs on a PC under the Windows 95 Operating System and supplies aircraft detection and aircraft flight information to WEBSTAR.

3.3.3 WEBSTAR Components

WEBSTAR can be viewed as being composed of two components, the Message Component and Air-Defence Component. The Message Component is for reporting on

DICE messages both during and after simulation, and sources its data from the DICE Ingres database. The Air-Defence Component is for reporting on air-defence related simulation data such as aircraft detections and aircraft missions, and sources its data from the ADSIM data file EVENTS.DAT.

WEBSTAR can be executed in two environments:
1. Entirely on UNIX machines (Figure 12, Figure 15 and Figure 14 below);
2. The Message Component running on UNIX (Figure 12 and Figure 14) with the Air-Defence Component running on Windows 95 (Figure 13 and Figure 15).

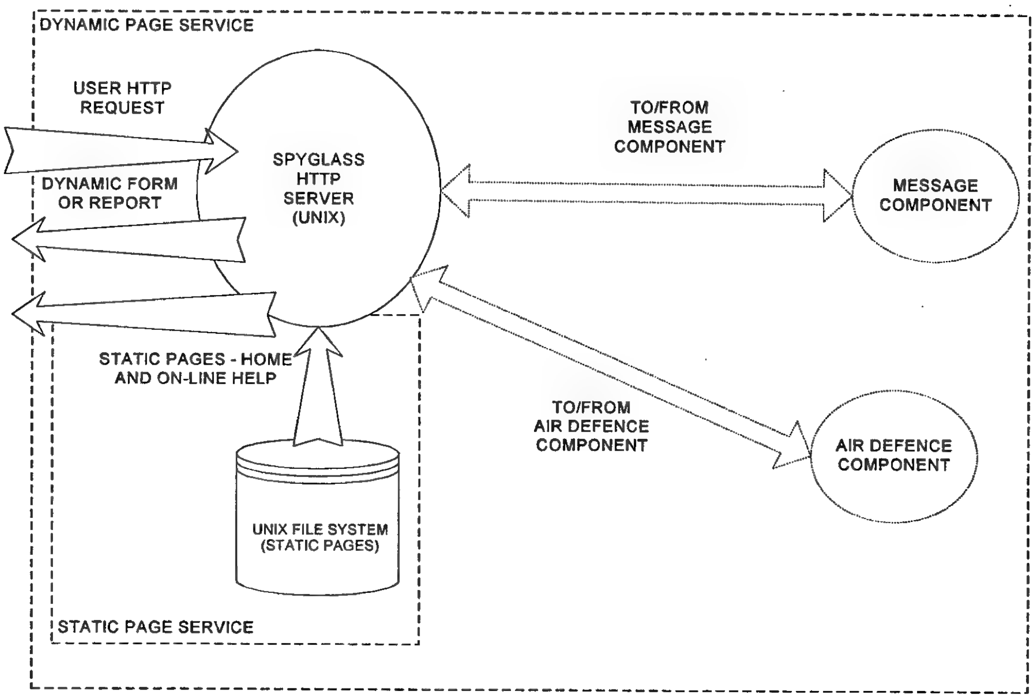

Figure 12 WEBSTAR on UNIX

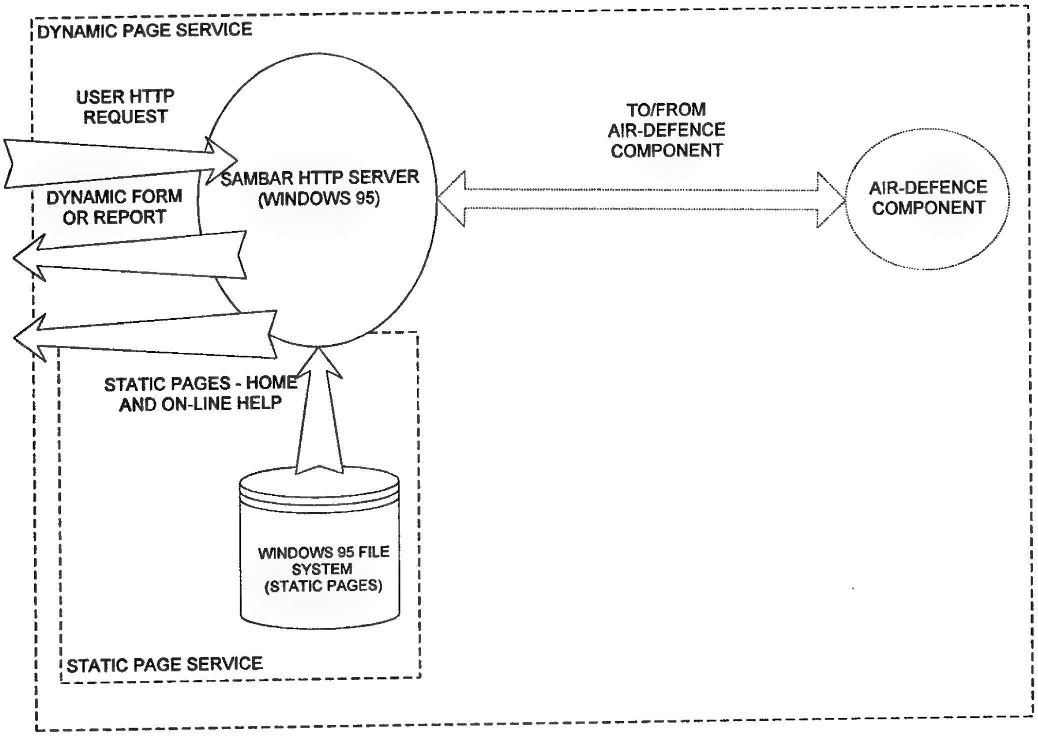

Figure 13 WEBSTAR on Windows 95

The former environment will require only one HTTP (Web) server running on a UNIX machine, whereas the latter environment will require two - one for UNIX and one for Windows 95. Hence this will involve the Air-Defence Component (Figure 15) linking with the extra, Windows 95, HTTP Server (Figure 13). The UNIX server and Windows 95 server will have separate Universal Resource Indicators (URI) [Berners-Lee 1995], because they will be running on different machines. For example, URIs `http://ssa-sun6:8081/` for the UNIX HTTP server and `http://itdpc75:8080/` for the Windows 95 HTTP server respectively.

In Figure 12, the CA-OpenIngres Spyglass HTTP Server [Computer Associates] will be running on a UNIX machine, waiting for user HTTP requests (via a web-browser on the network) for WEBSTAR pages.

In Figure 13, the Sambar HTTP Server [Sambar], will be running on a Windows 95 machine, waiting for the user's HTTP requests for pages.

User requests for pages may be serviced statically or dynamically (see section 3.3.1.1) depending on the page requested (defined by the URI). The home page and on-line help pages are all serviced statically (i.e. by responding with a static page stored in the file system). The WEBSTAR forms and reports are serviced dynamically, by retrieving

form or report data from the ADSIM data files or Ingres database, and incorporating this into an HTTP response, using HTML. Responses will also contain graphics (in compressed JPG or GIF formats). These are static and are requested by the browser.

3.3.3.1 *Message Component*

The Message Component (Figure 14) will always run on UNIX, since it must connect with the UNIX Ingres database. This database is updated by DICE components such as war games, command support systems, human players such as commanders, and peripheral units such as ADSIM. For more information on how this is done, see [Davies et al]. Embedded SQL statements are executed in the UNIX C programs for retrieving data from the Ingres database. This data is retrieved into C host language variables, for processing into the dynamically generated form or report. This form or report makes up the HTTP response to the user, via the web-browser.

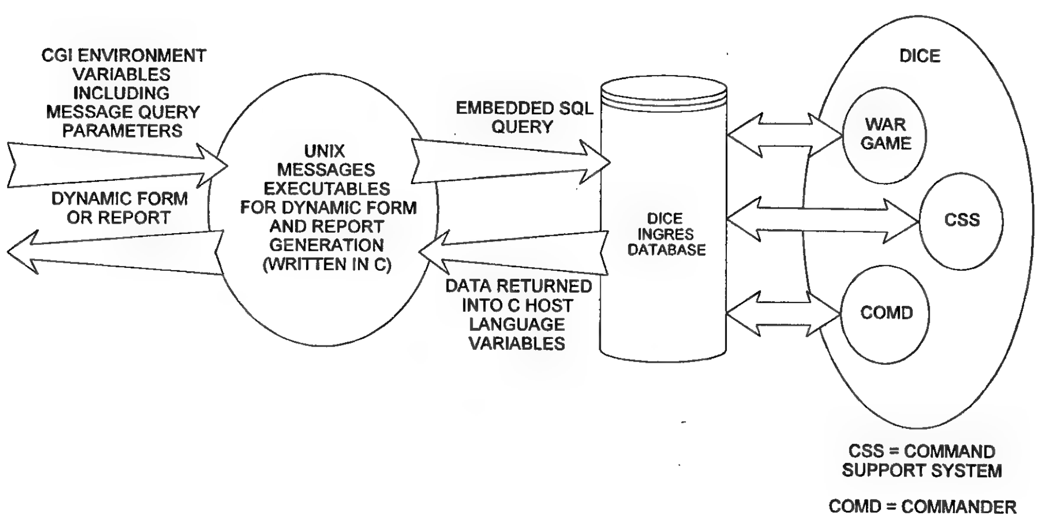

Figure 14 Message Component

3.3.3.2 *Air-Defence Component*

If the Air-Defence Component (Figure 15) is running on UNIX, then the relevant executables for dynamic form and report generation will be UNIX C executables containing code which reads the ADSIM data file EVENTS.DAT. Otherwise, these executables will be Microsoft Visual C++ executables running on a Windows 95 machine. The data files will be stored on a UNIX file system. In order to access these files from a PC, the path to the files must be mapped using the Windows Explorer Map Network Drive feature. For more information on how this is done, see [Haydon].

The ADSIM data files are used in the following way. The ADSIM Peripheral Unit Interface (PUI) writes to SPC.TXT in response to messages sent to ADSIM from other components of DICE. ADSIM reads SPC.TXT and takes actions based on its contents. ADSIM writes events to EVENTS.DAT, e.g. data such as aircraft detections. For a detailed discussion of how these files are updated and read, see [Haydon].

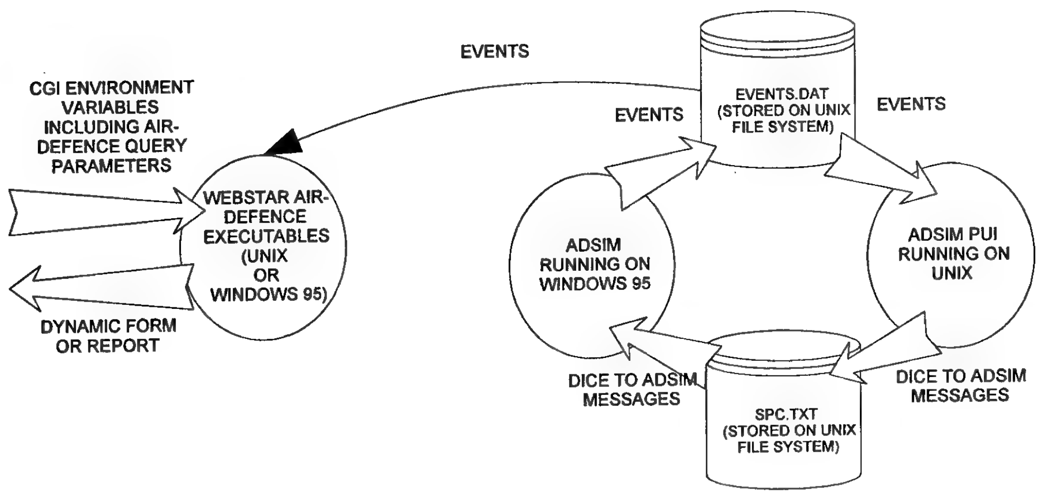

Figure 15 Air-Defence Component

4. Discussion of Web Technologies in C3I Simulation

4.1 Insights Gained into the Use of Web Technologies

The most important lessons learned whilst producing WEBSTAR were:

4.1.1 Application Development Time

Development of web applications involves understanding and integrating many technologies, such as HTML, HTTP and CGI. Tools for Rapid Application Development (RAD) of web applications are not established, unlike those that are well established for RAD Local Area Network (LAN) applications (e.g. Microsoft Visual Basic, Borland Delphi, CA-OpenIngres, Oracle). However, a range of tools are being released. At present, development of a web application will generally take longer than an equivalent application written for a LAN.

4.1.2 Application Installation Time

Installation and maintenance of client web application software is simple. If a web-browser is installed on the WEBSTAR user's machine, and a connection to the

WEBSTAR web server is available, no installation is required. In addition, if the WEBSTAR application needs to be maintained, it only needs to be changed on the WEBSTAR web server, and all changes are effective for all users (with perhaps a click of the 'Reload' button in the web-browser necessary). The browser serves as a front end for multiple databases (i.e. the DICE message database and ADSIM log files).

4.1.3 Code Libraries

In web application development, there exists, like in any application development, scope for code re-use. Libraries written for retrieving and processing CGI parameters from HTML forms, and for outputting well-formed HTML markup will decrease future development effort.

4.1.4 Dynamically and Statically Generated Web Pages

Dynamic forms and reports can be developed. These are created by CGI programs which use some data source such as a simulation log file or database to determine the content of the form or report. The term dynamic is used to signify that HTML is created 'on-the-fly' by the CGI program, rather than being stored in a static file. The dynamically generated HTML contains data taken from the processing of the data source. Dynamic forms increase the flexibility of the application to adapt to different simulation scenarios.

4.1.5 Security

There are security issues involved with CGI and the CA-OpenIngres Spyglass server which need to be examined closely for the situations where the simulation involves classified data.

4.1.6 Network Accessibility

One advantage of using a web application rather than a traditional LAN application is greater network accessibility. The data can be accessed over a much larger network, ranging in size from a very large wide area network (WAN) (e.g. the Internet) to intranets, or local area networks (LAN).

However, in order to maintain realism within the simulation, it is sometimes necessary to restrict information available to simulation players (see section 3.2.4).

4.1.7 Scalability and Project Planning

If an application is developed initially as a web application on a LAN (an intranet application), then it is easy to scale across the Internet if required later (subject to hardware resource availability and response times across the Internet).

In contrast, if the application was developed initially as a LAN application, then it may be more difficult to later convert to a web application and scale across the Internet. Or the LAN application may need to be replaced by a completely new web application, resulting in greater costs than if it was initially developed as a web application.

If the application was not originally intended to work across the Internet (e.g. a legacy mainframe application), then the web application could not have been planned for and the cost of either converting this into a web application or writing a completely new application is additional to the cost of the legacy application.

In addition, according to the comments made in section 4.1.1, web application development tools are not as mature as LAN application development tools. It generally costs more to develop a web application than an equivalent LAN application. The respective development times and methods are however converging, with many products being released for web application development.

4.1.8 Pros and Cons of Using HTML Forms/CGI

HTML forms coupled with CGI is one solution for web-based user interfaces. Java and Javascript are two others, but during this project, there was not sufficient time to use these. However, they should be investigated.

The advantages of using HTML forms/CGI are :
- One web client (e.g. Netscape) can serve as a front end for multiple databases (e.g. for the ADSIM data files and DICE database). With WEBSTAR, all forms and reports are displayed in the browser window;
- One database can talk to multiple clients on different platforms (e.g. UNIX Netscape and Windows Netscape), each with its native platform interface characteristics;
- Changing the user interface does not require changing all clients in the field - only the user interface pages accessed by clients (assuming that the most recent version of each page is always loaded by the browser, i.e. no viewing of outdated, cached pages);
- The display of form controls is easy;

The costs and limitations of using HTML forms/CGI are :
- The interface does not support field formatting (input masks) or an exhaustive set of data types;
- Dynamic updating of form controls depending on user selection in other controls is not supported;
- There is little control over the positioning of controls;
- It does not support client-side user input validation;
- It requires the user to press a submit button for any server involvement (and hence all input validation is done by the server);
- Navigation among various input fields can be awkward on some platforms. There are no keyboard shortcuts and tabbing between fields may not work as it normally does on a particular platform;

- HTTP is a stateless protocol. There is no concept of state in a user session using CGI and HTTP, although attempts to get around have been made (e.g. cookies).

4.2 Examples of Potential Simulations of ADF Messages and Reports

It may be feasible to simulate, within DICE, standard ADF messages and reports by retrieving data from DICE and presenting it in an appropriate format. For example, rather than use an HTML intelligence report (INTREP) published by a human player, data could be retrieved from the simulation, and either listed in tabular format, or converted into some natural language (i.e. English) text.

These messages and reports would be accessed by interested parties using a web browser. An archive of previous web pages containing the messages and reports could be maintained.

Examples would be :

- INTREP (intelligence report)/INTSUM (intelligence summary) ;
- SITREP (situation reports) - for access by other units;
- fuel states/weapon states of aircraft. This could be modelled by Petri Nets [Bowden] which include code for updating web pages;
- LOG/PERS REPLEN (logistics/personnel replenishment) information (e.g. when the next water replen will occur). This could be modelled by Petri Nets [Bowden];
- communication equipment status (COMMREPs). This could be simulated by interrogating existing files in ADSIM.

Example of web pages published by human players could be:

- orders;
- concept of operations;
- high-level plans;
- rules of engagement;
- VIP visit information;
- weather reports and forecasts.

5. Recommendations

In order to decrease future development effort, code libraries should be developed from the code used written in WEBSTAR, for retrieving and processing CGI parameters from HTML forms, and for outputting well-formed HTML markup.

The following recommendation is a result of the observations made in section 4.1. If there is great uncertainty before application prototype development begins as to whether the prototype will be successfully demonstrated and hence developed into a full web application across the Internet, then perhaps the lowest risk strategy would be

to use LAN RAD tools to develop the prototype. That way, if full development is rejected, then less cost will have been incurred than if the prototype had been developed as a web application. However, if it is likely that the prototype will be developed into a full web application across the Internet, then a web application prototype should be developed, then scaled up across the Internet.

The following recommendations are aimed at further research and development into web technologies in C3I simulation, especially WEBSTAR and DICE.

1. In order to maximise effectiveness of using various communications technologies, generic research should be conducted into selecting the most appropriate communications technology for particular tasks (e.g. is e-mail or web publishing most appropriate for the information being communicated?). Based on this research, write a report on the appropriate use of web publishing and browsing;
2. Use the research presented herein into technologies to drive development of web browsing capabilities in DICE and to interoperate with JCSE.
3. Incrementally develop more components to WEBSTAR (e.g. aircraft mission details);
4. Perform algorithmic complexity analysis of WEBSTAR programs and thorough testing of WEBSTAR within many DICE scenarios.
5. Allow sorting of WEBSTAR report records other than in chronological order of simulation time;
6. Investigate web security;
7. Investigate Java and Javascript for re-engineering of WEBSTAR;
8. Investigate Dynamic HTML (both Microsoft and Netscape definitions);
9. Develop a Web Peripheral Unit Interface (PUI) - a DICE PUI which receives messages from other agents and publishes HTML pages depending on the message contents;
10. Incorporating ADFORMS->Text conversion into the Human Player (HP) facility, and subseqently incorporating this into the WEBSTAR message report;
11. Adding a graphical facility to view the DICE simulation network;
12. Investigate automatic simulation of standard ADF reports e.g. INTREP, MISREP with data retrieved from the simulation;
13. Add ability of Human Player to publish own HTML pages easily.

6. Conclusions

An investigation was undertaken into the use of web technologies in C3I simulation, using DICE and WEBSTAR as a working environment. A working web application, a set of Web-based Simulation Tools for Analysis and Reporting (WEBSTAR) was developed. WEBSTAR is now in use within the DICE environment. Insights into the issues associated with the use of web technologies have been gained from the development of WEBSTAR and additional research. This has provided valuable information for future web technology research and development in DICE.

7. Acknowledgements

The author would like to thank, in no particular order, Noel Haydon for his tremendous help and advice on ADSIM, Carsten Gabrisch for advice on DICE, UNIX and Ingres, and Thomas Schar for advice on ADFORMS and C. Thanks must also go to Chris Dabovich for drawing diagrams, and Michael Gorroick, David Fogg, Stephen Hood and Michael Davies for facilitating my development of WEBSTAR. I would also like to thank Michael Davies for his advice on DICE, JCSE, and C3I simulation.

8. References

Berners-Lee, T., Connolly, D., "Hypertext Markup Language - 2.0", RFC 1866, MIT/W3C, November 1995
<URL:http://www.cis.ohio-state.edu/htbin/rfc/rfc1866.html>

Borenstein, N., and N. Freed. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993
<URL:http://www.cis.ohio-state.edu/htbin/rfc/rfc1521.html>

Computer Associates International, Inc., "CA-OpenIngres DocuROM", Computer Associates International, Inc., 1995

Cooper, A., "About Face - The Essentials of User Interface Design", IDG Books Worldwide, Inc., August 1995

Date, C.J., "An Introduction to Database Systems (Fifth Edition)", Addison-Wesley, 1995

Davies, M., Gabrisch, C., Dunn, J.M., Bowden, F.D.J., "The Distributed Interactive C3I Effectiveness (DICE) Simulation", DSTO Technical Report, DSTO-TR-0485, February 1997

Grobe, M., "Instantaneous Introduction to CGI Scripts and HTML Forms", The University of Kansas, July 1996
<URL:http://kuhttp.cc.ukans.edu/info/forms/forms-intro.html>

Haydon, N.A. (1996), "Establishing an Interface to a DOS-based Air-Picture Simulation via a Local Area Network", DSTO General Document, DSTO-GD-0117, November 1996

Sambar, T., "Sambar Server Documentation", Sambar Technologies, 1997
<URL:http://www.sambar.com/docs/index.htm>

Van Biesbrouck, M., "CGI Security Tutorial", University of Waterloo Computer Science Club
<URL:http://www.csclub.uwaterloo.ca/u/mlvanbie/cgisec/intent.html#auto_1>

Bowden, D.J. (1996), "Petri Nets and Their Application to Command and Control Systems", DSTO Technical Report, DSTO-TR-0462, December 1996

An Investigation into the Use of Web Publishing and Browsing in C3I Simulation

Andrew Pope

(DSTO-GD-0181)

DISTRIBUTION LIST

		Number of copies
AUSTRALIA		
DEFENCE ORGANISATION		
Task sponsor:		
Director General Force Development (Joint) (Attention: Director of Information Systems)		1
S&T Program		
Chief Defence Scientist	)	
FAS Science Policy	)	1 shared copy
AS Science Corporate Management	)	
Director General Science Policy Development		1
Counsellor, Defence Science, London		Doc Control Sheet
Counsellor, Defence Science, Washington		Doc Control Sheet
Scientific Adviser to MRDC Thailand		Doc Control Sheet
Director General Scientific Advisers and Trials	)	1 shared copy
Scientific Adviser - Policy and Command	)	
Navy Scientific Adviser		1 copy of Doc Control Sheet and 1 distribution list
Scientific Adviser - Army		Doc Control Sheet and 1 distribution list
Air Force Scientific Adviser		1
Director Trials		1
Aeronautical & Maritime Research Laboratory		
Director		1
Electronics and Surveillance Research Laboratory		
Director		1
Chief Information Technology Division		1
Research Leader Command & Control and Intelligence Systems		1
Research Leader Military Computing Systems		1
Research Leader Command, Control and Communications		1
Executive Officer, Information Technology Division		Doc Control Sheet
Head, Information Architectures Group		1
Head, Information Warfare Studies Group		Doc Control Sheet
Head, Software Systems Engineering Group		Doc Control Sheet

Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Advanced Computer Capabilities Group	Doc Control Sheet
Head, Computer Systems Architecture Group	Doc Control Sheet
Head, Systems Simulation and Assessment Group	1
Head, Intelligence Systems Group	Doc Control Sheet
Head, CCIS Interoperability Lab	Doc Control Sheet
Head Command Support Systems Group	1
Head, C3I Operational Analysis Group	1
Head Information Management and Fusion Group	Doc Control Sheet
Head, Human Systems Integration Group	Doc Control Sheet
Head, C2 Australian Theatre	Doc Control Sheet
Head, Year 2000 Project	Doc Control Sheet
Author (Andrew Pope)	1
Publications and Publicity Officer, ITD	1
DSTO Library and Archives	
Library Fishermens Bend	1
Library Maribyrnong	1
Library Salisbury	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
Capability Development Division	
Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Director General C3I Development	Doc Control Sheet
Intelligence Program	
DGSTA Defence Intelligence Organisation	1
Corporate Support Program (libraries)	
OIC TRS Defence Regional Library, Canberra	1
Officer in Charge, Document Exchange Centre (DEC),	1
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1
Universities and Colleges	
Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Senior Librarian, Hargrave Library, Monash University	1
Librarian, Flinders University	1

Other Organisations	
NASA (Canberra)	1
AGPS	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations	
INSPEC: Acquisitions Section Institution of Electrical Engineers	1
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts	1
Documents Librarian, The Center for Research Libraries, US	1
Information Exchange Agreement Partners	
Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES 10

Total number of copies: 60

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
An Investigation into the Use of Web Publishing and Browsing in C3I Simulation	Document (U) Title (U) Abstract (U)
4. AUTHOR(S) Andrew Pope	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-GD-0181	AR-010-549	General Document	May 1998

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
N9505/15/83	ADF 96/210	D.O.I.S.D.	42	9

13. DOWNGRADING/DELIMITING INSTRUCTIONS	14. RELEASE AUTHORITY
N/A	Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

1. World Wide Web
2. Command, Control, Communications and Intelligence
3. Computerized Simulation
4. DICE Simulation

19. ABSTRACT
With the advent of World Wide Web technology, and in particular web browsing, and its increasing use in Australian command support systems such as the Joint Command Support Environment (JCSE), it is essential that a capability within C3I (command, control, communications and intelligence) simulations such as the Distributed Interactive C3I Effectiveness (DICE) Simulation be developed. Over a period of three months, a suite of tools for reporting on simulation data, using a web browser as the basis for the user-interface, was developed. The end product is called WEBSTAR (WEB-based Simulation Tools for Analysis and Reporting). WEBSTAR is now in use within the DICE environment. A set of criteria for evaluating web applications is suggested. Insights into the issues associated with the use of web technologies in C3I simulation have been gained from the development of WEBSTAR and additional research. This has provided valuable information for future web technology research and development in DICE.